

## Laboratorio 2 ejercicios y reporte

```
#include <iostream>
using namespace std;

void fun_a(int *px, int *py);
void fun_b(int a[], int tam);

int main()
{
    int array[10] = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9};
    fun_b(array, 10);

    void fun_a(int *px, int *py){
        int tmp = *px;
        *px = *py;
        *py = tmp;
    }

    void fun_b(int a[], int tam){
        int f, l;
        int *b = a;
        for (f = 0, l = tam - 1; f < l; f++, l--) {
            fun_a(&b[f], &b[l]);
        }
    }
}
```

Cuál es su dirección en memoria? ¿Cuántos bytes se dedican para almacenar cada elemento de array?

R://

El arreglo de A[10] usa 40 bytes de memoria , 4 bytes por cada elemento  
la posición de memoria que apunta al array es 0x7ffdc7583fc0

Cuál es la dirección y el contenido en memoria del elemento array[3] ?

R://

la posición de memoria en A[3] = 0x7ffdc7583fcc

Describe el efecto que tiene la función fun\_b, sobre el arreglo array.

R://

está realizando un ordenamiento (invertir el array)  
así:

```
0 1 2 3 4 5 6 7 8 9
9 1 2 3 4 5 6 7 8 0
9 8 2 3 4 5 6 7 1 0
9 8 7 3 4 5 6 2 1 0
9 8 7 6 4 5 3 2 1 0
```

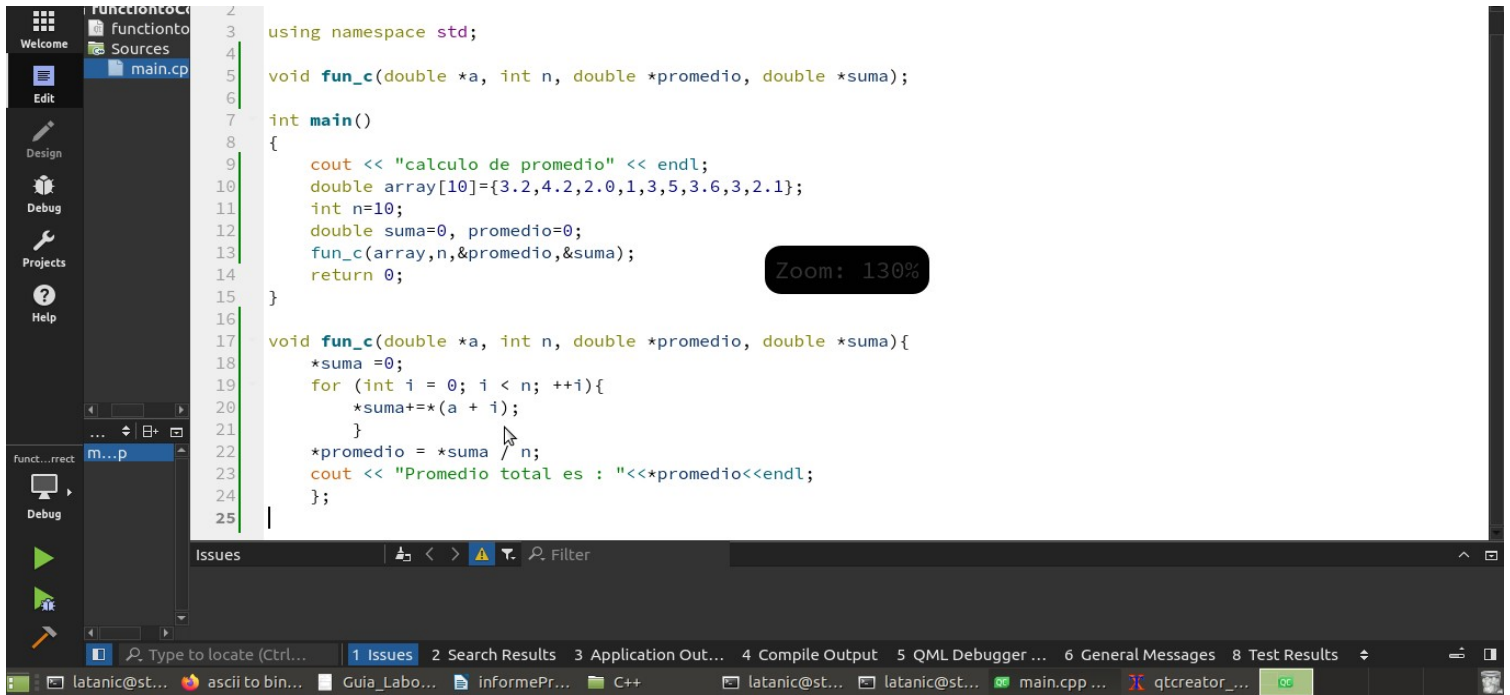
```
void fun_c(double *a, int n, double *promedio, double *suma){
    int i;
    suma = 0.0;
    for (i = 0; i < n; i++)
        suma += (a + i);

    promedio = suma / n;
}
```

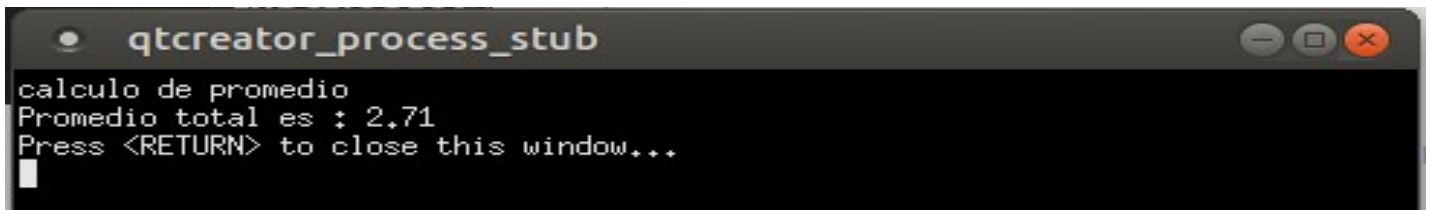
Corregir la función que se presenta en la imagen anterior

R://

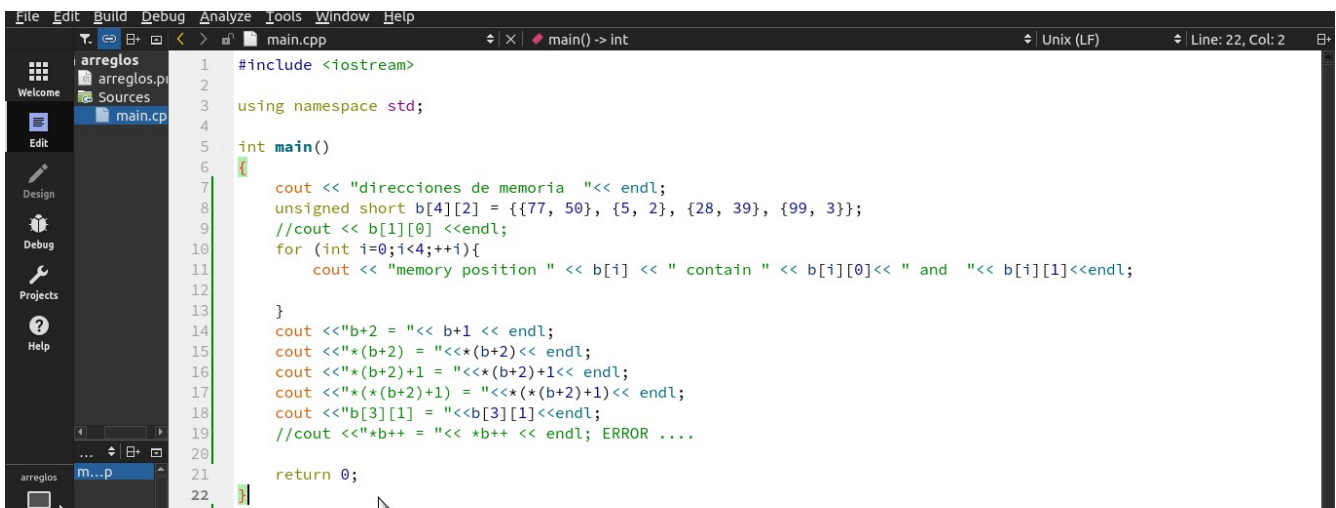
Corrección de la función



Resultado de consola



MEMORY



### Resultado de la consola

```
qtcreator_process_stub
direcciones de memoria
memory position 0x7ffd1ef2e7d0 contain 77 and 50
memory position 0x7ffd1ef2e7d4 contain 5 and 2
memory position 0x7ffd1ef2e7d8 contain 28 and 39
memory position 0x7ffd1ef2e7dc contain 99 and 3
b+2 = 0x7ffd1ef2e7d4
*(b+2) = 0x7ffd1ef2e7d8
*(b+2)+1 = 0x7ffd1ef2e7da
*(*(b+2)+1) = 39
b[3][1] = 3
Press <RETURN> to close this window...
█
```